

REMARKS**Status of the Claims**

Claims 1, 3-8, 10-14, and 16-20 are currently present in the Application, and claims 1, 8, and 14 are independent claims. Claims 1, 3-8, 10-14, and 16-20 have been amended, claims 2, 9, and 15 have been canceled, and no claims have been added.

Applicants are not conceding in this Application that those canceled claims are not patentable over the art cited by the Examiner, as the present claim amendments and cancellations are only for facilitating expeditious prosecution of the present Application. Applicants respectfully reserve the right to pursue these and other claims in one or more continuations and/or divisional patent applications.

Examiner Interview

Applicants note with appreciation the telephonic interview conducted between Applicants' representative and the Examiner on October 17, 2007. During the telephonic interview, the Examiner and Applicants' representative discussed one of the 103 references (Ansari et al., U.S. Patent No. 6,473,897). In particular, Applicants' representative discussed that Applicants' invention translates byte code into processor-specific object code **at runtime using a runtime loader** based upon a computer system's environment. In contrast, Ansari identifies a processor type **during compilation** and merely executes processor-specific code at runtime. The Examiner suggested that describing that a runtime loader performs particular tasks at runtime would position Applicants' claims to read over the art of record. Such amendments are included in this response.

Drawings

Applicants note that the Examiner did not indicate whether the formal drawings, filed with Applicants' application, are accepted by the Examiner. Applicants respectfully request that the Examiner indicate whether the formal drawings are accepted in the next office communication.

Information Disclosure Statement

Applicants note that the Examiner did not initial reference AN in Applicants' IDS filed on March 29, 2007, and also did not initial reference AP in Applicants' IDS filed May 20, 2007. Applicants request the Examiner consider these references and send a copy of initialed form PTO-1449 for each IDS to Applicants.

Claim Rejections – Alleged Obviousness Under 35 U.S.C. § 103

Claims 1-4, 6-7, 14-17, and 19-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ansari et al. (U.S. Patent No. 6,473,897, hereinafter "Ansari") in view of Kempf et al. (U.S. Patent No. 5,359,721, hereinafter "Kempf"). Applicants respectfully traverse these rejections. Claims 2 and 15 have been canceled in this response and, therefore rejections to these claims are moot.

Applicants have amended independent claim 1 to distinctly claim that Applicants invention compiles code into byte code, and then uses a runtime loader during runtime to identify a processor type and translate the byte code into object code that executes on the identified processor type. Support for this amendment may be found in Applicants' specification on page 47, line 24 through page 50, line 4. Therefore, no new matter is added with such amendment. As amended, claim 1 is directed toward a method with limitations comprising:

- compiling source code, the compiling resulting in byte code;
- retrieving the byte code at runtime using a runtime loader; in response to retrieving the byte code, using the runtime loader to identify a processor type from a plurality of heterogeneous processor types to execute the byte code;
- in response to identifying the processor type, using the runtime loader to translate the byte code to object code; and
- loading the object code into a processor that corresponds to the identified processor type.

Applicants' invention uses a compiler to generate byte code from source code and, at runtime, Applicants' invention uses a runtime loader to translate the byte code to

processor-specific object code. In contrast, Ansari compiles processor-specific code from source code during compilation. Ansari states:

“...the compiler automatically and dynamically **analyzes the source code and customizes the object code** for particular processor types” (col. 11, lines 39-41)

“A computer-implemented method **analyzes a source code segment which is to be compiled for execution** by any one of several different processor types.” (Abstract)

As can be seen from the above excerpts, Ansari’s processor-type identification occurs during code compilation. During runtime, Ansari merely executes the object code. As a result, Ansari does not teach or suggest “*retrieving the **byte code at runtime using a runtime loader; ... using the runtime loader to identify a processor type** from a plurality of heterogeneous processor types to execute the byte code;*” and “*...using the runtime loader to translate the byte code to object code*” as claimed by Applicants. The Office Action does not suggest that Kempf teaches or suggests any of Applicants’ compilation or translation limitations, and indeed Kempf does not teach or suggest such limitations.

Therefore, since neither Ansari nor Kempf teach or suggest all of the limitations included in Applicants’ claim 1 as amended, amended claim 1 is allowable over Ansari in view of Kempf. Claim 14 is a computer program product claim including similar limitations as claim 1 and, therefore, is allowable for at least the same reasons that claim 1 is allowable as discussed above.

Notwithstanding the fact that claim 6 is dependent upon claim 1 and, therefore, allowable for at least the same reasons that claim 1 is allowable, claim 6 adds limitations to claim 1 of:

- detecting, using the runtime loader at runtime, one or more operations included in the byte code; and
- matching, using the runtime loader at runtime, one or more of the operations with one of the processor types from the plurality of heterogeneous processor types.

Applicants use the runtime loader at runtime to analyze the byte code and identify operations that correspond with a particular processor type. As discussed above, neither Ansari nor Kempf identify a processor type at runtime using a runtime loader, but rather identify a processor type during source code compilation. Therefore, neither Ansari nor Kempf teach or suggest “*detecting, using the runtime loader at runtime, one or more operations included in the byte code*”; and “*matching, using the runtime loader at runtime, one or more of the operations with one of the processor types from the plurality of heterogeneous processor types*” as claimed by Applicants. As such, since neither Ansari nor Kempf teach or suggest all of the limitations included in Applicants’ claim 6, claim 6 is allowable over Ansari in view of Kempf. Claim 19 is a computer program product claim including similar limitations as claim 6 and, therefore, is allowable for at least the same reasons that claim 6 is allowable as discussed above.

Notwithstanding the fact that claim 7 is dependent upon claim 1 and, therefore, allowable for at least the same reasons that claim 1 is allowable, claim 7 adds limitations to claim 1 of:

- determining, using the runtime loader at runtime, whether the byte code includes a program directive corresponding to one of the plurality of processors; and
- matching, using the runtime loader at runtime, one or more of the operations with one of the processor types from the plurality of heterogeneous processor types in response to the determination.

Applicants use the runtime loader at runtime to analyze the byte code and identify a program directive that corresponds with a particular processor type. As discussed above, neither Ansari nor Kempf identify a processor type at runtime using a runtime loader. Therefore, neither Ansari nor Kempf teach or suggest “*determining, using the runtime loader at runtime, whether the byte code includes a program directive corresponding to one of the plurality of processors; and matching, using the runtime loader at runtime, one or more of the operations with one of the processor types from the plurality of heterogeneous processor types in response to the determination*” as claimed by Applicants. As such, since neither Ansari nor Kempf teach or suggest all of the limitations included in Applicants’ claim 7, claim 7 is allowable

over Ansari in view of Kempf. Claim 20 is a computer program product claim including similar limitations as claim 7 and, therefore, is allowable for at least the same reasons that claim 7 is allowable as discussed above.

Each of claims 3-4 and 16-17 depend upon one of the allowable independent claims 1 or 14. Therefore, each of claims 3-4 and 16-17 are allowable for at least the same reasons that their respective independent claims are allowable.

Claims 5, 8-13, and 18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ansari in view of Kempf and further in view of Jacobson (U.S. Patent Pub. 2003/0188045, hereinafter "Jacobson"). Applicants respectfully traverse these rejections. Claim 9 has been canceled in this response and, therefore, rejection to this claim is moot.

Notwithstanding the fact that claim 5 is dependent upon claim 1 and, therefore, allowable for at least the same reasons that claim 1 is allowable, claim 5 adds limitations to claim 1 of:

- wherein the identifying includes using the runtime loader to analyze the availability of each of the plurality of heterogeneous processor types, and wherein the analyzing includes retrieving a loading factor for each of the plurality of heterogeneous processor types, which corresponds to the availability of each of the plurality of heterogeneous processor types.

Applicants' determine processor-loading factors during a time at which processors are executing (runtime). The combination of the prior art would not function properly because Ansari only teaches identifying a processor type at code compilation, which is at time at which the system is not yet executing code and, therefore, the loading factor for all processors would be zero. Therefore, combining the prior art would not yield a reasonable expectation of success. As such, per MPEP 2143, the Examiner has not established a prima facie case of obviousness and, therefore, claim 5 is allowable over Ansari in view of Kempf. Claim 12 is an information handling system claim including similar limitations as claim 5 and, therefore, is allowable for at least the same reasons that claim 5 is allowable as discussed above. Claim 18 is a computer

program product claim including similar limitations as claim 5 and, therefore, is allowable for at least the same reasons that claim 5 is allowable as discussed above.

Claim 8 is an information handling system claim including similar limitations as claim 1. The Office Action does not suggest that Jacobson teaches or suggests the limitations in claim 8 that are similar to claim 1, and indeed Jacobson does not teach such limitations. Therefore, claim 8 is allowable over Ansari in view of Kempf in further view of Jacobson for at least the same reasons that claim 1 is allowable over Ansari in view of Kempf.

Each of claims 10-13 depend upon independent claim 8 and, therefore, are allowable for at least the same reasons that claim 8 is allowable.

Conclusion

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Leslie A. Van Leeuwen, Reg. No. 42,196/
Leslie A. Van Leeuwen, Reg. No. 42,196
Van Leeuwen & Van Leeuwen
Attorney for Applicant
Telephone: (512) 301-6738
Facsimile: (512) 301-6742